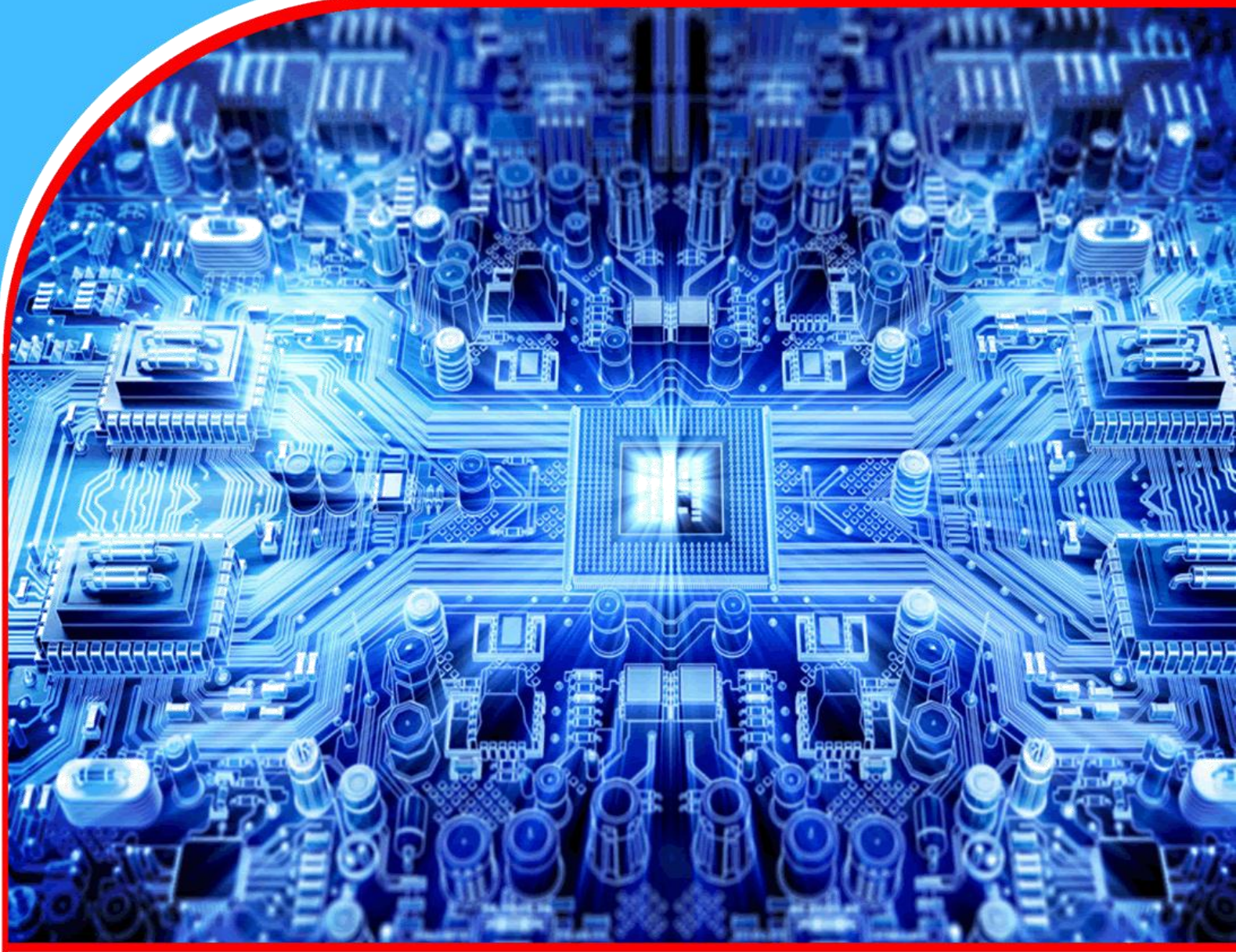
American Journal of
Computing and Engineering
(AJCE)



Design and Development of a VHF Telemetry Two-way Radio Device Network Downtime/Uptime Alerting Prototype

Jonathan Chisapi and Dr. Charles S Lubobya

American Journal of Computing and Engineering
ISSN 2790-5586 (Online)
Vol.5, Issue 1, pp 36 - 57, 2022
www.ajpojournals.org

Design and Development of a VHF Telemetry Two-way Radio Device Network Downtime/Uptime Alerting Prototype

[1*]Jonathan Chisapi and [2]Dr. Charles S Lubobya

[1,2]Electrical Department, School of Engineering University of Zambia.

*Corresponding Author's Email: chisapijonathan@yahoo.com

ABSTRACT

This paper presents the design and development of a VHF telemetry Two-way radio device network downtime/uptime alerting prototype. Telemetry network monitoring using VHF 2-way radio Streaming is a paradigm in networking management that has not been deployed. The framework enables continuous monitoring of the mobile and fixed access points of the mining fleet management Wi-Fi system (in the case of the Barrick Lumwana mine, Zambia). Considering the significance of real-time alerts in a mining facility, critical systems such as slope monitoring radars, mobile emergency alerts, and timely copper production reports must be monitored in real-time. The results for the performance warnings of the telemetry access points collectively accounted for a response alert time of 1999ms from the samples collected which translated to 0.29% compared to the SNMP system that registered a total of 689500ms for the samples collected which translates to 99.71% of the total response time of 691499ms. The results derived from the VHF 2-way radio telemetry prototype show that this system performs better than the traditional SNMP network monitoring system. We recommend that this IP network prototype is used in open-pit mines alongside Motorola telemetry two-way radios.

Keywords: *Network Telemetry, Simple network monitoring protocol (SNMP), Very high frequency (VHF)*

1. INTRODUCTION

The acquisition of telemetry data is becoming increasingly important for efficient detection and timely reaction to changes in network state, such as Wi-Fi device uptime/downtime alerts. The current traditional simple network monitoring protocol, (SNMP), uses the pull-based data collection mechanism. On average, network operators use SNMP data every 5 to 30 minutes, which affects the visibility of the network status [1]. Telemetry network using VHF 2-way radio utilizes the idea of "push the data" not "pull the data". The VHF 2-way radio network telemetry prototype harnesses and delivers near-real-time device alert status (uptime/downtime).

Comparing network telemetry with SNMP, to retrieve any information from a network device using SNMP, NMS (Network Management System) needs to first request this data in form of an SNMP request. Only then can the data be sent from the network device back to the NMS in form of SNMP response message/s. This is repeated every polling interval. To retrieve a considerably large amount of data, SNMP polling relies on GetBulk operations. It performs a continuous GetNext operation that retrieves all the columns of a given table (e.g., *ifTable, a unique value greater than zero*). The network device will return as many columns from the *ifTable* as can fit into a single packet. If the polling NMS detects that the end of the table has not yet been reached, it will do another GetBulk and will repeat the operation until the whole *ifTable* is fetched [2]

Streaming network telemetry gains efficiency over SNMP by eliminating the polling process. Instead of sending SNMP requests with specific instructions which the network device process every time, telemetry uses a configured policy on the device which allows you to know which data to collect, how often and to which NMS should be sent. This paper focuses on building and comparing a VHF 2-way radio network telemetry system with SNMP polling. The results are then populated for both systems.

2. NETWORK MONITORING METHODS

There are different kinds of network monitoring methods. Some of the methods covered below are as follows: Simple Network Management Protocol (SNMP), OpenFlow-based, and Programmable Data Plane-based.

2.1. Simple Network Management Protocol (SNMP)

For many years, Simple Network Management Protocol has been considered the benchmark for monitoring enterprise networks. Recently SNMP has long been plagued by a host of vulnerabilities, which include limited data recovery and filtering options, unreliable transmission, and inconsistent encryption between versions [1].

SNMP is used in network management systems to monitor devices connected to a network for conditions that require administrator attention. SNMP presents management data as variables on the managed system. These variables can then be queried (and sometimes set) by application management. SNMP itself does not define which variables are accessible; Instead, SNMP uses an extensible design in which the available information is determined by management information

bases (MIBs) that are typically owned by individual vendors [1]. The MIB delineates the management data structure of a device subsystem in a hierarchical namespace containing an object identifier (OID). Each OID identifies variables that can be read or set via SNMP. The main principle of SNMP is based on simple operations that allow network administrators to query and establish the status of certain devices. Although SNMP is capable of managing a wide variety of network devices like printers, personal computers, servers, power supplies, etc., it is mainly associated with routers and other network devices [1].

2.1.1. SNMP architecture

SNMP consists of a manager and an agent. The architecture is a database of management information, managed objects, and the network itself. They all work together, as the manager provides the interface between the human interface and the management system. The physical devices include systems like servers, routers, or hubs. These objects are arranged in MIB. The MIB is a Management Information Base known as virtual information database. SNMP permits managers and agents to communicate, to access these objects in the MIB.

There are five significant messages which are used by the SNMP to communicate between the manager and the agent. These messages include Get, GetNext, GetResponse, Set, and Trap. The manager requests information for specific variables using the GET or GetNext, message. Once the agent receives a Get or GetNext message, it issues a GetResponse message to the manager with the requested information. If it encounters an error, it explains the reason for the error caused in processing. The trap message allows the agent to inform the manager of a 'significant' event. Figure 1 below shows the SNMP architecture.

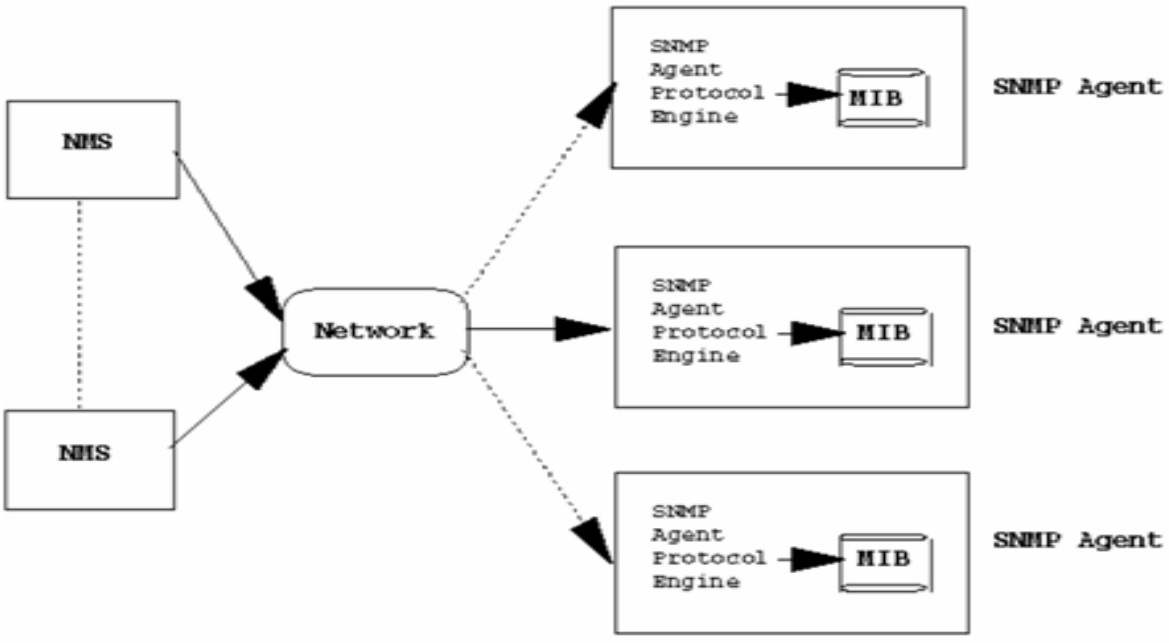


Figure 1: SNMP Architecture

The SNMP manager issues all the messages such as Get, GetNex,t, and Set. Only the Trap message is capable of being initiated by the agent. It is used by RTUs (Remote Telemetry Units (RTUs) to report alarms. This immediately notifies the SNMP manager when an alarm condition occurs, instead of waiting for a response from the SNMP manager.

2.2. Telemetry

2.2.1. Wireless telemetry

Telemetry is a highly automated communication process in which measurements and other data are collected at remote or inaccessible points and transmitted to receiving devices for monitoring, viewing, and recording [2]. Modern telemetry often uses radio transmissions. The main applications of telemetry are monitoring power plants, collecting meteorological data, and monitoring manned and unmanned flights [3].

Figure 2 shows a typical telemetry system that consists of an input device called a transducer, a transmission medium (usually radio waves), signal receiving and processing equipment, and recording or displaying equipment[4].

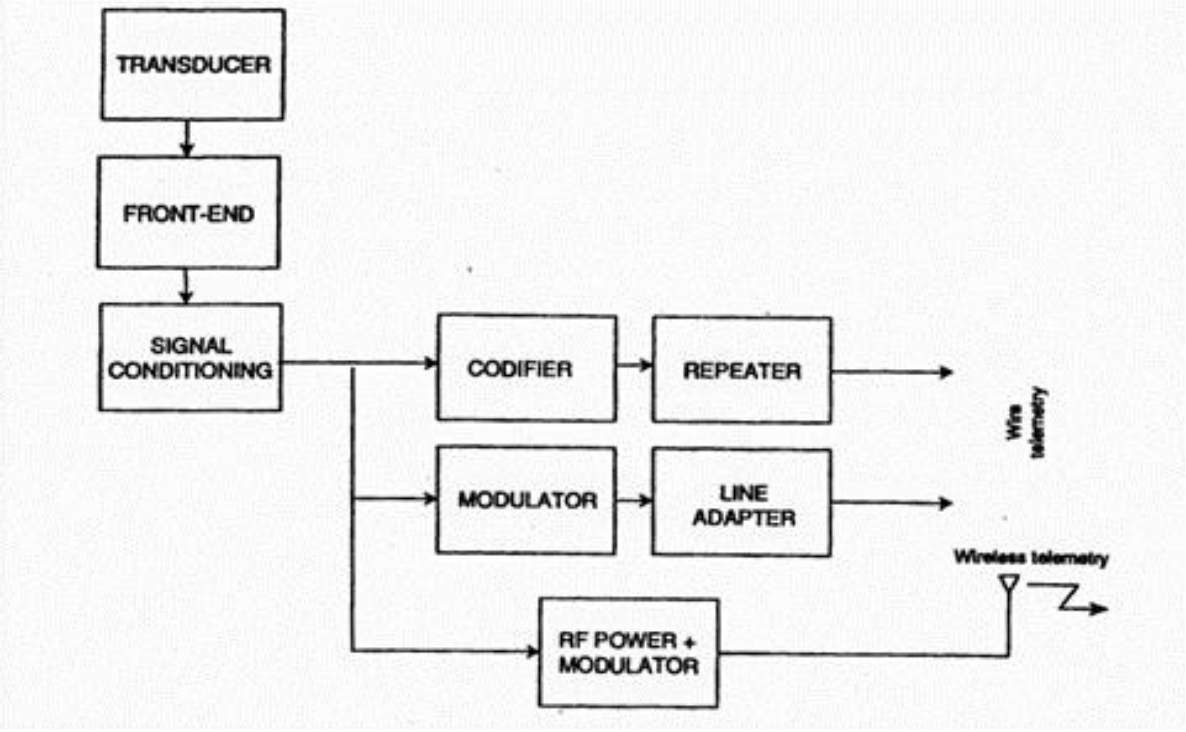


Figure 2: Block diagram for a telemetry system

For wireless telemetry, these modules are a receiving antenna designed for maximum efficiency in the RF band used; a radio receiver having a demodulation scheme compatible with the modulation scheme, and demodulating circuitry for each of the transmitted channels[4]. For wired telemetry, the antenna and radio receiver are replaced with a generic front end to amplify

the signal and match the line impedance to the input impedance of the following circuits. The transmission in telemetry systems, especially wireless ones, is done by transmitting a signal whose analog variations in amplitude or frequency are a known function of the variations in the signals coming from the transducers [5].

2.2.2. sflow and Netflow network monitoring system

Traditionally, there are many different monitoring techniques used in computer networks [6]. Although two traditional methods, NetFlow and sFlow [7], have been widely used for many years. NetFlow performs per-flow monitoring, which obtains information from the IP stream as it passes through the switch, then exports the aggregated data. Therefore, NetFlow needs additional memory and CPU to extract and process stream data. Furthermore**,** a monitoring center is required to perform polling on the switches to obtain data. Since the shortest period for exporting NetFlow data is 15 seconds, hence NetFlow is not suitable for real-time monitoring. SFlow on the other hand samples the flow of packets passing through a network interface. sflow takes a sample once every (n) packet, with a configurable sampling rate (n.) The sample packet maybe then sent immediately to the target instance for further analysis. sFlow requires a little extra CPU and memory on the switch but has a downside in terms of accuracy. The sampling method may not detect microbursts and anomalies and may also not detect small flows. . Figure 3 below shows the sflow and netflow network monitoring system.

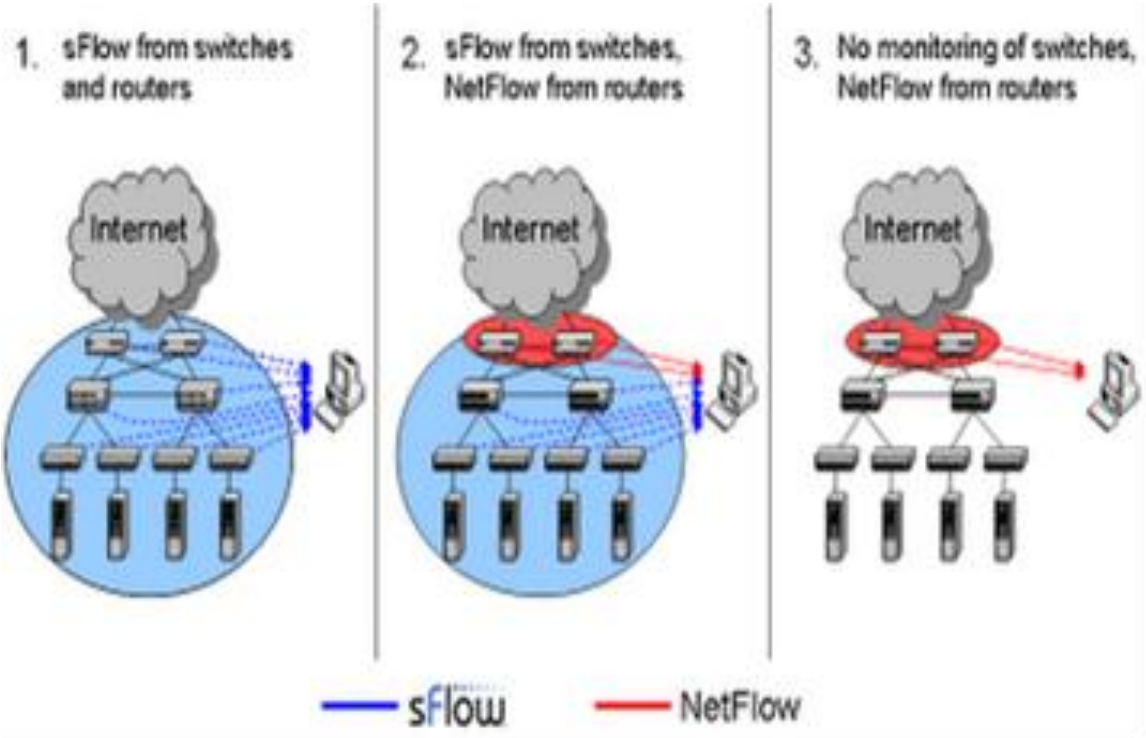


Figure 3: sflow and Netflow network monitoring system

2.3. OpenFlow-based

By separating the control and data planes, a new method has been developed for monitoring in SDN environments. Most of these are based on OpenFlow, de facto standard SDN protocol. FlowSense [8] proposed a push-based monitoring approach rather than a polling-based approach. FlowSense uses Packet-In, and Flow-Removed messages sent from switches to controllers to report network status. It has very low overheads at the cost of not being able to perform accurate real-time monitoring. OpenNetMon [9] uses a polling technique instead. The polling rate is adaptive, can be increased when the range changes quickly, or decreased when it is stable to minimize the number of requests. Adaptive polling provides fair accuracy while maintaining low CPU overhead. However, it does not provide information about the intermediate switch and does not have a complete picture of the state of the network, because the Open NetMon poll only monitors the edge switches. Open Sample [10] is another monitoring framework for the Open Flow protocol that uses the sFlows packet sampling approach. Therefore, the purpose of Open Sample is traffic engineering, as it can quickly detect elephant flows and estimate link usage.

2.4. Programmable Data Plane-based

Many pieces of research on programmable data planes have been proposed. Moving switches from fixed functions to programmable functions. Some new methods have been proposed to take advantage of this capability, such as Open Sketch [11], Univ Mon[12], Insitu OAM (iOAM) [13],and INT. Open Sketch and UnivMon work with sketch-based streaming algorithms in switches, enabling accurate monitoring with low overhead. OpenSketch is implemented in FPGA switches while UnivMon is deployed using P4. iOAM and INT are specifications for integrating network telemetry data into user traffic.

3. RELATED WORK

Many studies have been conducted on network monitoring. Most of them were based on wired networks because it is simple to obtain network traffic statistics in real-time, or at least in near real-time [3]. The problem with these studies is that they cannot be readily applied to wireless networks. To obtain the same level of detail in wireless networks as in wired networks, the previously noted challenges need to be addressed. Recently solutions have been proposed in academia as well as industry for network monitoring. However, existing solutions mainly concentrate on trade-offs between expressiveness, accuracy, speed, and scalability [1], [5]. For instance, systems like NetQRE [7] and others can support a wide range of queries using stream processors running on general-purpose CPUs, but they incur substantial bandwidth and processing costs to do so. Telemetry systems such Chimera [6] and Gigascope [8] are expressive by covering a wide range of telemetry items, however, can only support lower packet rates. This is due to the systems to process all packets at the stream processor which can become a bottleneck.

Roughan [14] made a case study on the accuracy of Simple Network Management Protocol measurements of link loads in the Abilene network. In their paper, they stated that SNMP allows

the collection of data such as the number of bits, and packets to cross an interface over a time interval, typically every five minutes. In some research works of literature, it is sometimes seen as a poor source of data compared to others such as flow-level collection. This belittlement towards SNMP arises in parts because it often suffers from artifacts, errors, and missing data. Nevertheless, it is hard to overestimate how useful SNMP data is to network managers, and how much work has gone into extending this utility by making it possible to use this data to estimate traffic matrices [15] and detect anomalies [16]. Hence, it is significant to consider the accuracy of SNMP measurements.

Other previous studies have considered errors in SNMP link-load measurements as being based on arguments unsupported by data [14]. For instance, Zhao et al. [17] argue that SNMP errors are primarily caused by errors in polling timestamps, and present an error model, but data is not used to validate the model or assumption about the cause of errors. Few works have considered the accuracy of the measurements themselves, one exception being [2], which looks at packet loss measurements.

2.5. CWRADAR and FAP08 HIA

The access point prototypes used and the naming conversions are as follows: CWRADAR stands for chimi-west RADAR (Radio detection and ranging) while FAP08 HIA stands for fixed access point number 8 located in the mine high industrial area. The main active mining pit is called the chimi main pit area. Figure 4 and figure 5 below show CWRADAR and FAP08 HIA access points respectively.

Figure 4: CW RADAR Access point

Figure 5: FAP08 HIA Access point

4. SYSTEM COMPONENTS

There are a variety of components we used to set up our system that was categorized as follows: access point, network monitoring, and software components. We built our project using the following equipment listed below.

4.1. Access point components

- Battery, solar application 12 – 102 AMP Enertec
- Regulator, solar power 12 – 24 V DC 20A Phocos
- Solar panel 140W – 12V Polycrystalline
- DC power injector 9 – 36 VDC
- CAT 6 Ethernet cable
- 16A circuit breaker Schneider
- Antenna, 2.4GHZ WIFI 9DBI Omnidirectional
- 2.4GHZ Ubiquiti M2 bullet radio
- Whip, mobile aerial 2DB VHF
- RG58 coaxial cable
- SRD – 12 VDC – SL – C Relay
- Radio, base digital-analog Motorola DM4400

4.2. Network monitoring system components

- Arduino Uno board R3 (Microcontroller)
- Radio, base digital-analog Motorola DM4400
- Whip, mobile aerial 2DB VHF
- Monitoring windows 10 PC

4.3. Software requirements

- Arduino IDE Development environment
- C++ for the microcontroller programming.
- The dude 3.6
- MOTOTRBO V13.0 programming software

4.4. Physical setup of Network Access points

The two prototype access points were each installed with the system as highlighted in section 6. The main objective of the designed and built system was to stream and push network statistics

(uptime/downtime power status alerts) from the network device in real-time. An integrated system of the VHF Motorola 2-way radio and Ubiquiti wifi bullet radio were set up on each project access point as shown in figure 6.

Figure 6: An integrated system of the VHF Motorola 2-way radio and Ubiquiti wifi bullet radio.

5. TOPOLOGY

Two different topologies were designed, one for the SNMP network and one showing the VHF layout.

5.1. SNMP

The pit Wi-Fi network helps sustain and manage the mine's mineral hauling fleet and other related systems. The topology was designed for the two access points under study. The dump trucks and other auxiliary equipment are installed with a fleet management modular system that aids in automatic assignments i.e. fuel assignments, dump and load assignments. All equipment is linked to the control room or OEM simulation room via the fixed and mobile access points. The 2.4 GHz frequency is used on the access points and the mining equipment (fleet and other systems) while the 5.8 GHz link is used to haul traffic from the tower to the main dispatch control room. Figure 7 depicts the SNMP network topology. The Wi-Fi design is similar to the work done in [18]

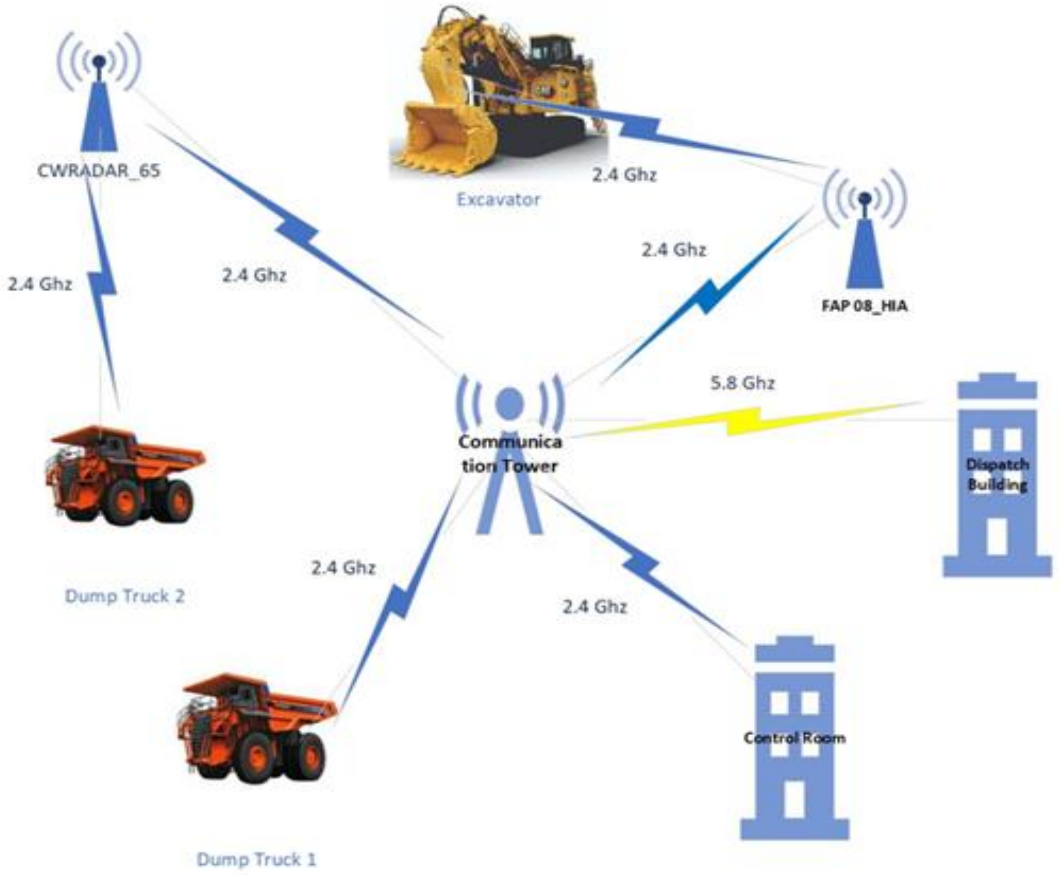


Figure 7: SNMP Wi-Fi Topology

5.2. VHF Radio Layout

Three Radios, base digital-analog Motorola DM4400 were set up on FAP0, CWRADAR access points (alongside the SNMP system), and the simulation control room. All the three radios used some simplex channels for communication. The frequency allocated to these radios for this project was 136.000MHZ for both transmit and receive. Figure 8 below shows the integrated VHF 2-way radio system layout.

American Journal of Computing and Engineering
ISSN 2790-5586 (Online)
Vol.5, Issue 1, pp 36 - 57, 2022
www.ajpojournals.org

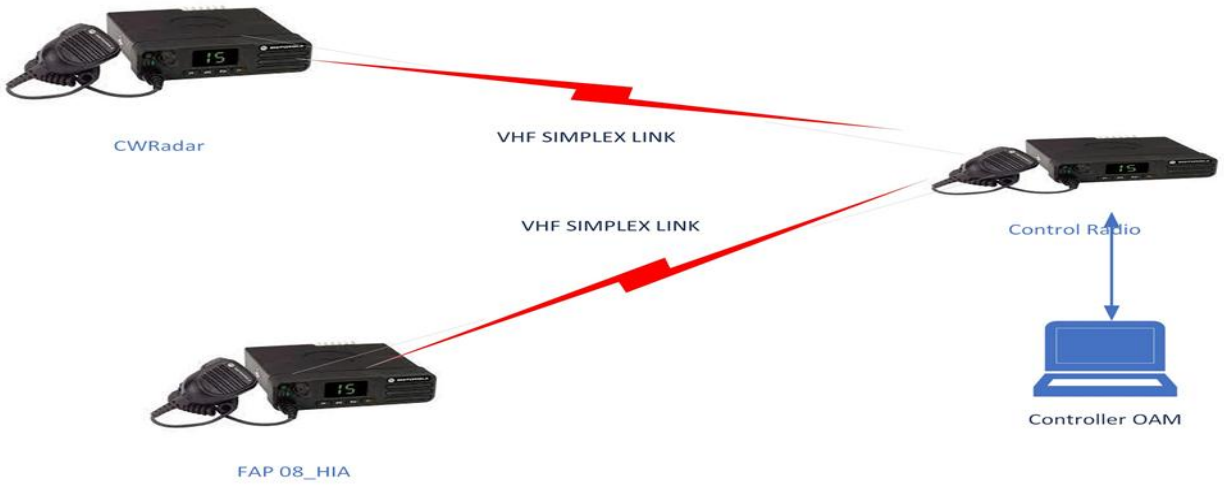


Figure 8: VHF 2-Way Radio Layout

6. SYSTEM IMPLEMENTATION

In the implementation of telemetry with VHF 2-way radio, the setup was done as follows: For each of the two prototype access points, one 12v solar application battery, one Solar panel, 140W – 12V Polycrystalline were connected to the regulator solar power 12 – 24 V DC 20A Phocos (One regulator, solar power 12 – 24 V DC 20A Phocos was used for each system). For both simulations, 16A circuit breakers Schneider were connected from the solar regulator to the solar panel, to the battery and the loads. Connecting to the loads, the Ubiquiti Wi-Fi radio was biased through the POE while the VHF radio was connected to the relay. An SRD – 12 VDC – SL – C Relay was linked and wired with the 12v switched battery from the radio rear accessory pin 7 which is SW+ and pin 8, PWRGND negative. Figure 9 below shows a DM4000 series radio rear accessory pin layout.

Figure 9: VHF 2-Way Radio rear accessory

The other side of the relay module in figure 10 below, has pins – a Ground pin and a VCC pin to power the module and an input pin IN to control the relay. On the VCC terminal, a 12VDC (pin7) source is connected while a 5V level GPIO from the radio rear pin is connected to an input pin IN to control the relay.

Figure 10: SRD – 12 VDC – SL – C Relay

The SRD – 12 VDC – SL – C Relay operates in a normally closed setup, this means that the electromagnetic part of the relay is continuously activated by the 12VDC. When this happens, a 5V level GPIO signal is transmitted via VHF whip antenna to the control radio that is linked to an Arduino circuit as shown in figure 11 below.

Figure 11: Monitoring room setup

6.1. Network Monitoring

6.1.1. VHF Telemetry

On the network monitoring and software part, we used an Arduino Uno board R3 (Microcontroller) and a Radio, base digital-analog Motorola DM4400 for control. The telemetry function allows you to monitor and control the GPIO status of one radio from another radio. When an external device is connected to a radio, you can monitor that external device through a control radio. Moreover, the remote radio can detect the running status change of the external device via the level change of the GPIO pin on a real-time basis. In this case, external devices connected to both the control radio and the monitored radio are an Arduino and relay respectively.

Figure 12 below shows how we linked the Arduino Uno board with both the monitoring computer and the 2-way radio.

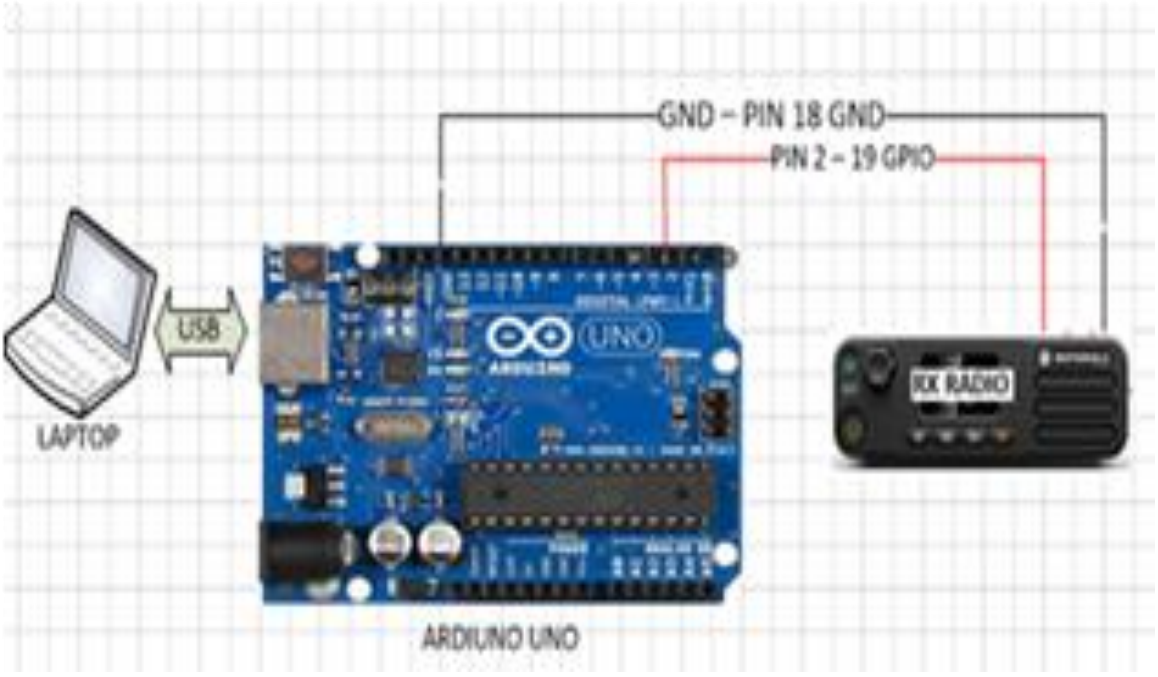


Figure 12: Detailed Ardiuno Uno Connections

The Computer graphically displays the data as it is collected. The program that was used in the experiment is the C++ programming language installed on a laptop with the Windows 10 operating system, it allowed the exchange of data between an Arduino board and a computer.

6.1.2. SNMP

The 2.4GHZ Ubiquiti bullet radio was configured and set up in the monitoring room so that it connected to the LAN port of the laptop. The dude server was installed on the laptop, dude server is an application that runs in the background. The Dude uses a Laptop as a client. In addition to

the client/host, client is also used as a Dude server that controls and monitors the LAN network.

After a dude software was installed on a laptop, a Windows server 2016 SNMP monitoring in Dude Services is configured. Both the CWRADAR (chimi west radar) and FAP08 HIA (fixed access point high industrial area) were linked to the main communication tower wirelessly while the simulation control room had a physical LAN connection from the tower. Figure 13 below shows the dude version 3.6 software monitoring screen. The green color denotes that the access point is on while the red color shows the access point is off.

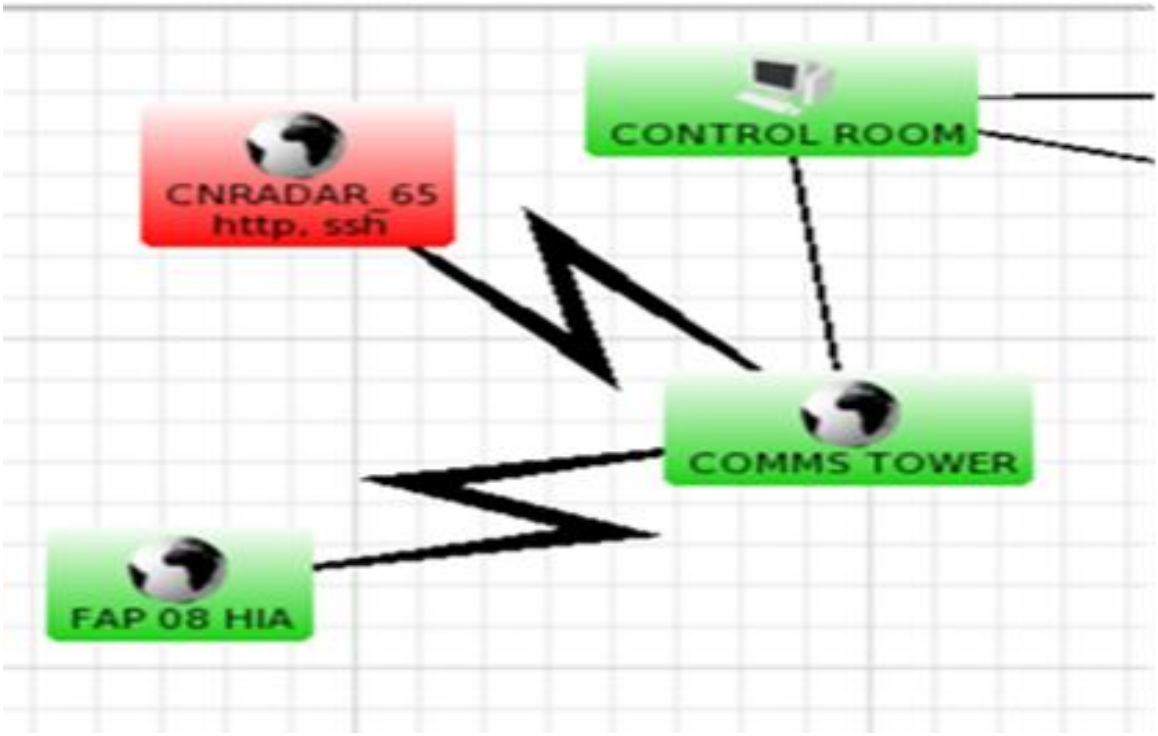


Figure 13: SNMP Network Monitoring Application

7. RESULTS AND DATA ANALYS

In this section, the results of the measurements described are presented. Results from the measurements related to VHF telemetry are presented in Section 7.1, followed by those related to SNMP in Section 7.2. The results for both solutions are presented in the order defined by the comparison criteria which in turn were presented in Section 8.

7.1. Telemetry

Below are some of the results obtained from the experiment for both the CWRADAR_65 and FAP 08 HIA access points' prototype simulations. CWRADAR and FAP08 HIA access points were both sampled randomly.

7.1.1. CW Radar_65

Table 1 below shows the test samples that were carried, the time, name of the access point, status, and the average time. Five alert response times were collected for CWRADAR_65. The average of the response times was derived from these recordings and further analyzed in section 7.1.2.

Table 1: CW Radar_65

Sample	Time	Name	Status	Response	Avg.
1	16:34:01.173	CWRADAR	OFF	422	211
	16:34:01.595		ON		
2	16:34:10.501	CWRADAR	ON	431	215
	16:34:10.923		OFF		
3	16:34:11.345	CWRADAR	OFF	375	188
	16:34:11.720		ON		
4	16:34:33.235	CWRADAR	ON	375	188
	16:34:33.610		OFF		
5	16:34:34.829	CWRADAR	OFF	422	211
	16:34:35.251		ON		

7.1.2. CW Radar_65 Analysis

Figure 14 below shows the five different test counts that were carried. Sample counts 1, 2, 3, 4, and 5 registered average alert response times of 211ms, 215ms, 188ms, 188ms, and 211ms obtained from table 1.

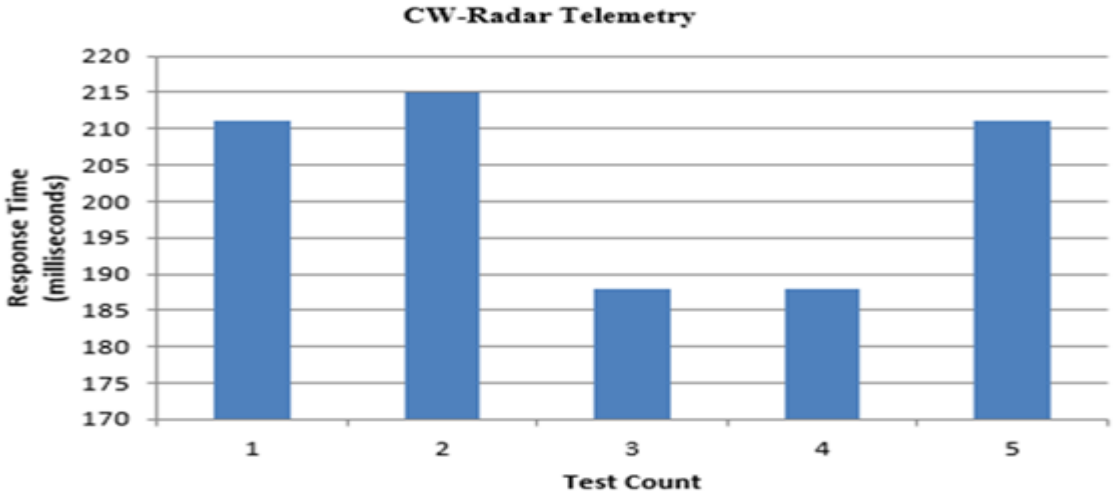


Figure 14: CW Radar_65 Analysis

Sample 1 and sample 5 had the same response time of 422ms with samples 3 and 4 also having the same response time of 375ms. Sample 2 registered an ON status time of 16:34:10:501 and an OFF status time of 16:34:10:923 having a response time of 431ms.

7.1.3. FAP08 HIA

Table 2 below shows five response time samples that were tabulated by getting the average of each sample. It also indicates the access point ON and OFF times, from which the response and average times are tabulated.

Table 2. FAP08 HIA

Sample	Time	Name	Status	Response	Avg.
1	16:26:26.028.	FAP08 HIA	OFF	375	188
	16:26:26.403		ON		
2	16:20:34.877	FAP08 HIA	ON	375	188
	16:20:35.252		OFF		
3	16:20:36.096	FAP08 HIA	ON	375	188
	16:20:36.471		OFF		
4	16:26:45.856	FAP08 HIA	ON	422	211
	16:26:46.278		OFF		
5	16:26:47.075	FAP08 HIA	OFF	422	211
	16:26:47.497		ON		

7.1.4 FAP 08 HIA Analysis

Figure 15 below shows recordings of both test counts and average response times collected from each sample. Counting the difference between the ON state and OFF state of the access point, samples numbers 1, 2, and 3 recorded 188ms while sampling numbers 4 and 5 registered averages of 211ms.

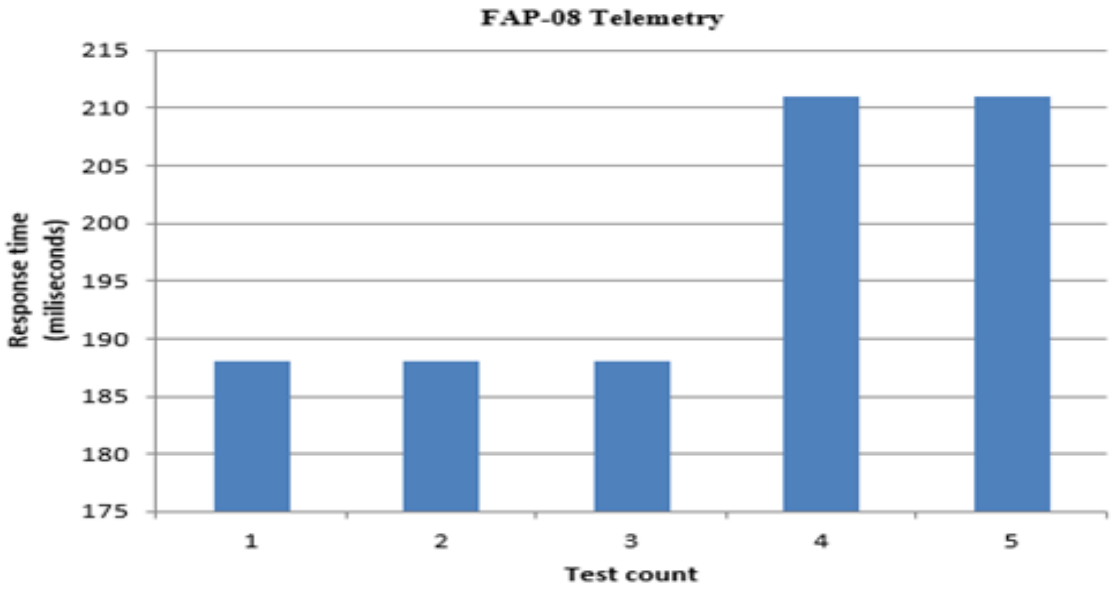


Figure 15: FAP 08 Telemetry

7.2. SNMP

Below are some of the results obtained from the experiment for both the CWRADAR_65 and FAP 08 HIA access points' SNMP simulations. Section 7.2.1 covers CW Rader while section 8.2.3 covers FAP08 HIA access points.

7.2.1. CW Radar_65

Table 3 presented in this section shows the response times, access point name, status, average, and response times.

Table 3: CW Radar

Sample	Time	Name	Status	Response	Avg.
1	15:20:29	CWRADAR	ON	267000	133500
	15:24:56		OFF		
2	15:27:29	CWRADAR	ON	23000	11500
	15:27:52		OFF		
3	15:31:59	CWRADAR	ON	105000	52500
	15:33:44		OFF		
4	15:51:53	CWRADAR	ON	34000	17000
	15:52:27		OFF		
5	16:43:30	CWRADAR	ON	50000	25000
	16:44:20		OFF		

7.2.2. CW Radar_65 Analysis

Figure 16 below records response and average times in milliseconds recorded from the five samples. Sample 1 recorded an average polling alert time of 133500ms, sample 2 registered 11500ms with sample numbers 2, 4, and five having 52500ms, 17000ms, and 25000ms respectively.

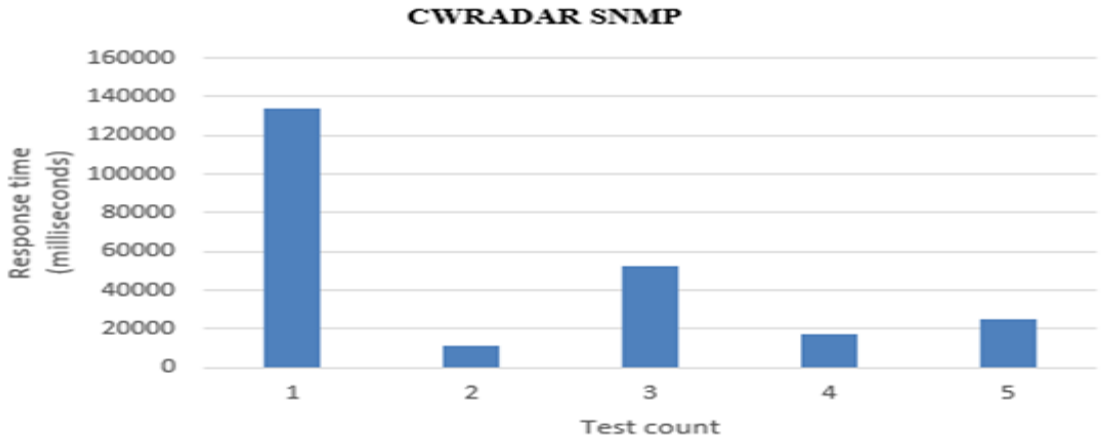


Figure 16: CWRADAR SNMP

7.2.3. FAP 08 HIA

Table 4 below sums up the SNMP response time sample recordings. A total of five samples for FAP 08 HIA access point response times were captured. Sample 1 recorded an ON time status of 13:24:10 and an OFF time of 13:25:59. Sample 2 recorded 13:31:29 ON and OFF time of 18:32:59 while sample number 3 had an ON time of 13:43:59 and an OFF status time of 13:47:29. The table below further indicates an ON time of 13:04:40 and an OFF time of 13:13:40 for sample number 4. Finally, sample number 5 indicates an ON status time of 16:15:00 and OFF status of 16:16:00. The access point analysis (FAP08 HIA) is presented in section number 8.2.4.

Table 4: FAP08 HIA

Sample	Time	Name	Status	Response	Avg.
1	13:24:10	FAP08 HIA	ON	109000	54500
	13:25:59		OFF		
2	13:31:29	FAP08 HIA	ON	90000	45000
	13:32:59		OFF		
3	13:43:59	FAP08 HIA	ON	210000	105000
	13:47:29		OFF		
4	13:04:40	FAP08 HIA	ON	540000	270000
	13:13:40		OFF		
5	16:15:00	FAP08 HIA	ON	60000	30000
	16:16:00		OFF		

7.2.4. FAP 08 HIA Analysis

All samples recorded from section 8.2.2 were analyzed as follows: Sample one recorded an average of 54500ms, sample two recorded 45000ms, sample three recorded 105000ms, sample four polled an average of 270000ms, and sample five registered 30000ms as shown in figure 17 below.

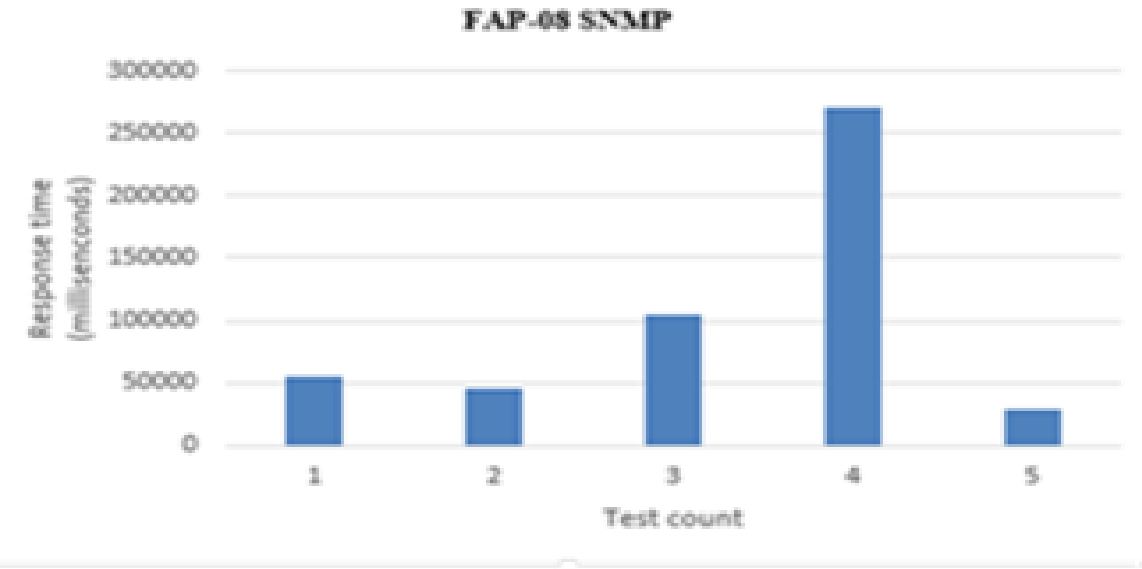


Figure 17: FAP08 SNMP

8. COMPARISON BETWEEN SNMP AND VHF 2-WAY RADIO TELEMETRY

Figure 18 shows a summarized response time alert comparison of the SNMP system and VHH 2-Way radio system. The first simulation percentage of VHF two-way radio telemetry CWRADAR recorded 0.15% and FAP 08 HIA telemetry registered 0.14%.

In the second Prototype simulation carried out, CWRADAR dude SNMP recorded 65.08% of response alert time while FAP 08 HIA dude SNMP records 34.63% of the response time. This translated a total of 1999ms for telemetry and a total of 744000ms for SNMP. The telemetry monitoring system further accounted for a system response alert time decrease of 34392.2%.

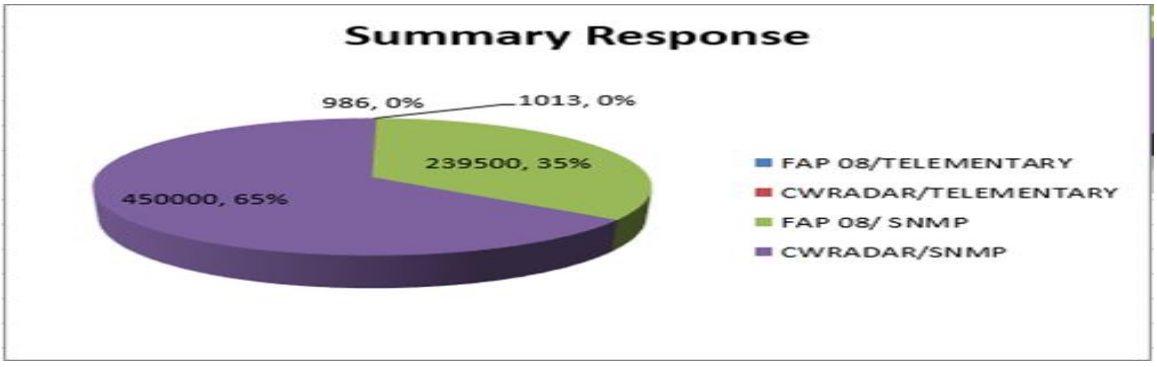


Figure 18: Summary Response Alerts

9. CONCLUSION

In this paper, a near real-time device Wi-Fi network uptime/downtime alerts using telemetry with VHF 2-way radio has been presented. According to the results in section 7, the telemetry prototype built in this study performed better as compared to the SNMP system.

The results for the performance warnings of the telemetry access points collectively accounted for a response alert time of 1999ms from the samples collected which translated to 0.29% compared to the SNMP system that registered a total of 689500ms for the samples collected which translates to 99.71% of the total response time of 691499ms.

The telemetry with VHF two-way radio system has shown to be more efficient in providing near real-time Wi-Fi network power alerts. The collection of real-time monitoring data helps to immediately evaluate and react quickly to the alerting current events.

10. RECOMMENDATIONS

The proposed project is a basic design, and it only demonstrates two access points. As for future works and recommendations, several areas were identified during the research process and the analysis of the results. The most prominent area where we recommend that this system is used is in open pit mines that use Motorola telemetry two-way radios alongside an IP network. Furthermore, future autonomous networks will require a holistic view on network visibility. The VHF monitoring project only accommodated two access points. Thus, more input terminal connections must be invented in order to make this network monitoring system robust. Factors that could as well be examined in that case are for example how the number of TCP connections affect the collector, and how memory and CPU use scales with the number of such connections.

ACKNOWLEDGMENT

I would like to thank Barrick, Lumwana copper mine Zambia and its operational technology department for granting me access to all resources I needed to complete my project. I also wish to extend my special wish to Mr. Paul Gillot, the mine manager for sponsoring my dissertation.

References

[1] R. B. Johnson, “EVALUATING THE USE OF SNMP AS A WIRELESS NETWORK MONITORING by,” no. May, 2009.

[2] T. Group *et al.*, “DOCUMENT 119-06 TELEMETRY APPLICATIONS HANDBOOK.”

[3] “Telemetry | communications | Britannica.” [Online]. Available: https://www.britannica.com/technology/telemetry. [Accessed: 23-Sep-2021].

[4] A. Lozano-nieto, “Albert Lozano-Nieto. ‘Telemetry.’ Copyright 2000 CRC Press LLC. <http://www.engnetbase.com>.,” 2000.

[5] H. K. Verma, “TELEMETRY SYETEMS.”

[6] N. L. M. Van Adrichem, C. Doerr, and F. A. Kuipers, “OpenNetMon : Network Monitoring in OpenFlow Software-Defined Networks,” no. May 2014, 2019, doi: 10.1109/NOMS.2014.6838228.

[7] J. Hyun, N. Van Tu, J. W. Hong, and A. Background, “Towards Knowledge-Defined

Networking using In-band Network Telemetry," *NOMS 2018 - 2018 IEEE/IFIP Netw. Oper. Manag. Symp.*, pp. 1–7, 2018.

[8] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "FlowSense: Monitoring Network Utilization with Zero Measurement Cost," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7799 LNCS, pp. 31–41, 2013, doi: 10.1007/978-3-642-36516-4_4.

[9] N. L. M. Van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: Network monitoring in OpenFlow software-defined networks," *IEEE/IFIP NOMS 2014 - IEEE/IFIP Netw. Oper. Manag. Symp. Manag. a Softw. Defin. World*, 2014, doi: 10.1109/NOMS.2014.6838228.

[10] J. Suh, T. T. Kwon, C. Dixon, W. Felter, and J. Carter, "OpenSample: A low-latency, sampling-based measurement platform for commodity SDN," *Proc. - Int. Conf. Distrib. Comput. Syst.*, pp. 228–237, Aug. 2014, doi: 10.1109/ICDCS.2014.31.

[11] L. Lu, R. Lewis, M. Hu, and R. Lin, "Design and Implementation of a Wireless Networked Water Level Control System," *J. Comput. Commun.*, vol. 03, no. 05, pp. 159–163, 2015, doi: 10.4236/JCC.2015.35020.

[12] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, "One sketch to rule them all: Rethinking network flow monitoring with UnivMon," *SIGCOMM 2016 - Proc. 2016 ACM Conf. Spec. Interes. Gr. Data Commun.*, pp. 101–114, Aug. 2016, doi: 10.1145/2934872.2934906.

[13] F. Brockners, "Next-gen Network Telemetry is Within Your Packets : In-band OAM," 2017.

[14] M. Roughan, "A Case Study of the Accuracy of SNMP Measurements," vol. 2010, 2010, doi: 10.1155/2010/812979.

[15] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads," 2003.

[16] M. Roughan, C. Lund, and D. Donoho, "An Information-Theoretic Approach to Traffic Matrix Estimation."

[17] Q. Zhao, Z. Ge, J. Wang, and J. Xu, "Robust traffic matrix estimation with imperfect information," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 133–144, Jun. 2006, doi: 10.1145/1140103.1140294.

[18] Smart. C. Lubobya, Mqhele. E. Dlodlo, Gerhard de Jager, "Performance Evaluation of the Wireless Tree Wi-Fi Video Surveillance System", 16th International Conference on Computer Modelling and Simulation, pp 510-515, 2014.